



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Irányítástechnika és Informatika Tanszék

Feladatellenőrző rendszer továbbfejlesztése

SZAKDOLGOZAT

Készítette

Nyíri Tamás

Konzulens

Dr. Szeberényi Imre

2017. november 14.



TANSZÉKVEZETŐ

SZAKDOLGOZAT FELADAT

Nyíri Tamás

mérnök informatikus hallgató részére

Feladatellenőrző rendszer továbbfejlesztése

A programozás oktatásában fontos szerepe van az önálló gyakorlásnak, feladatmegoldásnak, amit jelentős mértékben segít a gyors és pontos visszajelzés. Ez gépi eszközökkel jól támogatható, sőt olyan funkciókkal is kiegészíthető, melyek ellenőrzése hagyományos eszközökkel nehézkes. A tanszéken a korábbi szakdolgozatok és diplomatervek keretében elkészített Cporta/Jporta elnevezésű automatikus kiértékelést és ellenőrzést végző feladatkiértékelő rendszer nemcsak a feladatok ellenőrzését segíti, hanem az oktatás egyes adminisztrációs feladatait is egyszerűsíti.

Az újratervezett rendszer felépítésében és kialakításában lényegesen rugalmasabb elődjénél. Így lehetőség nyílik több olyan modul kialakítására is, ami a feladatok megfogalmazását és kiértékelését támogatja. A hallgató feladata a meglévő kiértékelő modulok továbbfejlesztése, új támogató funkciók megvalósítása, jogosultsági rendszer felülvizsgálata, valamint az adminisztrációs funkciók továbbfejlesztése.

Feladatok:

1. Ismerje meg és mutassa be jelenlegi rendszer felépítését és főbb moduljait!
2. Mérje fel és elemezze a rendszerrel kapcsolatos igényeket, melyek oktatói oldalról jelennek meg a feladatok kiírása, ellenőrzése és a tárgyak adminisztrációja terén!
3. Tegyen javaslatot új funkciók megvalósítására (pl. feltöltéskor közvetlen kapcsolat valamilyen verziókezelővel, plágiumkeresés, lefedettség- és egyéb tesztek, tesztek export/import funkciói, kari vezetés által generált újabb adminisztrációs feladatok támogatása, stb.)!
4. Az igények és lehetőségek figyelembevételével egészítse ki a rendszert, úgy hogy minél jobban támogassa az oktatók munkáját.
5. Készítse el a szükséges dokumentációkat!

Tanszéki konzulens: Dr. Szeberényi Imre

Budapest, 2017. október 6.

Dr. Kiss Bálint
egyetemi docens
tanszékvezető

HALLGATÓI NYILATKOZAT

Alulírott *Nyíri Tamás*, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző, cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózataán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2017. november 14.

Nyíri Tamás

hallgató

Tartalomjegyzék

Bevezető	7
Más automatikus kiértékelő rendszerek	7
Moodle	7
edX, Open edx	8
A szakdolgozat felépítése	9
1. A Jporta (5-10 oldal)	10
2. Hallgatók értékelései	11
2.1. Értékelési rendszer (2-3 oldal)	11
2.2. Dinamikus mezők (3-5 oldal)	11
2.2.1. Dinamikus mezők egymásra hivatkozása	11
3. Automatizált feladatkiértékelő modul	12
3.1. Megvalósult funkciók	12
3.1.1. Jogosultságok felülvizsgálata (2-3 oldal)	12
3.1.2. Kód lefedettség ellenőrzés (2-3 oldal)	12
3.1.3. Feladatok (tesztek) importálása és exportálása (2-3 oldal)	12
3.1.4. Feladatok csoportosítása (2 oldal)	12
3.2. További lehetőségek	12
3.2.1. Plágiumkeresés (1-2 oldal)	12
3.2.2. Verziókezelő támogatás (1-2 oldal)	12
Összefoglalás	13
Köszönetnyilvánítás	14

Ábrák jegyzéke	15
Irodalomjegyzék	16

Kivonat

A programozás korszerű oktatásában egyre inkább előtérbe kerül az önálló tanulás és gyakorlás, melynek célja a hallgatók motiválása a mélyebb tudás megszerzésére. Ezt a széleskörben elterjedt internet hozzáférés tette lehetővé, hiszen így bárhol és bármikor elérhetőek ezek a rendszerek mindenki számára.

Ennek támogatására jött létre a BME Irányítástechnika és Informatika Tanszékén először a Cporta, majd később annak utódja, a Jportaként ismert webes oktatást segítő rendszer. A portál feladata egyfelől az oktatók igényeit kielégítő adminisztrációs felület biztosítása, másfelől hogy lehetőséget teremtsen a hallgatók programozási feladatainak automatikus kiértékelésére és ellenőrzésére.

A minél nagyobb fokú megbízható automatizálás mind a hallgatóknak, mind az oktatóknak nagy segítséget biztosít. A hallgatók szinte azonnal értesülnek a feltöltött munkájuk esetleges hiányosságairól, hibáiról, mely megkönnyíti ezek javítását. Oktatói szempontból pedig nagy mértékben csökkenti a feladatok ellenőrzésére fordítandó munkát.

Szakdolgozatomban bemutatom a Jporta meglévő funkcionálisait, különös tekintettel az adminisztrációs részekre és az automatikus kiértékelésre. Megtervezem és implementálom azon funkciókat, melyek akár oktatói, akár hallgatói oldalról növelik a portál értékét. Végezetül pedig javaslatot teszek további fejlesztési lehetőségekre.

Abstract

Bevezető

A programozói ismeretek elsajátításához merőben más módszertanra van szükség, mint egy irodalmi vagy jogász pályán. Az előbbi előnye, hogy megfelelő háttér biztosításával nagyban javítható a tanulási görbe, azonban sajnos az egyetemi körülmények között nincs lehetőség minden halltóval személyesen foglalkozni, hiszen ez óriási többlet munkát róna az oktatókra. Emiatt a hallgatók eredményes, mégis időtakarékos támogatása érdekében egyre nagyobb törekvés indult az automatizálása felé. Ezen megoldások számos előnnyel rendelkeznek:

- Az értékeléshez nincs szükség a beadások letöltésére, azok saját számítógépen történő fordítására, futtatására, majd kiértékelésére.
- Azonnali visszajelzés a beadás sikerességéről a hallgatóknak.
- Egyéb beadáshoz kapcsolható metrikákkal is dolgozhatunk: futási idő, memóriahasználat, stb.
- Automatikus pontszámítás a számított metrikák alapján.
- Határidők automatikus kezelése.
- Kommunikációs felület a hallgató és oktató között.

Más automatikus kiértékelő rendszerek

Moodle

A Moodle[6] egy teljes körű eLearning rendszer, mely nyílt forráskódú GNU GPL [4] licenc alatt készült. A Moodle-nek mára több, mint 120 ezer felhasználója van, a világ 232 országában [7], ami a nyílt forráskód előnyeivel kombinálva óriási potenciált jelent.

A Moodle általános lehetőséget biztosít az online oktatáshoz. Egyszerűen hozhatunk létre benne kurzusokat, melyekre a jelentkezést akár korlátozhatjuk is. A kurzusokon belül témakörökre bonthatjuk a tananyagot, melynek formája sokféle lehet: pdf fájlok, videók, külső weboldalak, stb. A résztvevők elsajátított tudásának ellenőrzése sem marad el: a Moodle általános rendszert biztosít online tesztek készítésére is, különféle jelleggel, mint pl. több válaszlehetőségből helyes(ek) kiválasztása, egy soros vagy éppen hosszabb saját szavas válaszok. A legtöbb fajta tesztnél lehetőségünk van a helyes válaszok megadására, így a portál azonnal ki is értékeli a beadást, ennek köszönhetően pedig a felhasználó azonnal értesül az elért eredményéről.

Elterjedtségének köszönhetően számos közösségi fejlesztésű modul is készült hozzá, melyek közül találhatunk szép számmal a programozás oktatására fókuszolóakat is. Ilyen például a Virtual Programming Lab [8] [1], mely támogat forráskódszerkesztést a böngészőben, programok futtatását, ellenőrzését és plágium ellenőrzést is. Mindezt a felhasználók a már megszokott Moodle környezetben érhetik el a kényelmes használat érdekében.

edX, Open edX

Az edX [2] egy nonprofit online kezdeményezés, melynek alapítói a Harvard University és a Massachusetts Institute of Technology. Ennek segítségével egyetemi szintű kurzusokat tartanak világszerte közel 10 millió felhasználóval és több, mint ezer kurzussal [3].

Az edX rendszer nem csak egyszerű tananyagok megtekintésére biztosít lehetőséget, de videókat és egyéb feladatokat is találunk a kínált kurzusokban. Emellett egyszerűen kapcsolatba léphetünk másokkal, akik a kurzusunkat hallgatják, ha éppen segítségre lenne szükségünk. A kurzusok többnyire ingyenesek, de sok esetben van lehetőségünk fizetés ellenében igazolást kapnunk a sikeresen elvégzett kurzusról.

A portálon lehetőségünk van programozással kapcsolatos kurzusok felvételére is. Ezek keretében pedig online kód írásra, fordításra, futtatásra és a helyesség ellenőrzésére is van mód, amik nélkül az önálló tanulás nagyon nehézkes lenne. Webes kódolást némely kurzusnál Codeboard¹ integrációval van megoldva, melynek köszönhetően nem is kell a tanuláshoz a megfelelő környezetet telepítenünk a saját gépünkre.

Az Open edX egy nyílt forráskódú platform, melyet az edX fejlesztett ki és tett szabadon

¹A Codeboard (<http://codeboard.io>) egy böngésző alapú fejlesztő környezet a programozás oktatás segítésére. Támogatja az edX, moodle, coursera és egyéb platformokat.

hozzáférhetővé saját oktatási környezetek létrehozására. Közel a teljes szerver oldali kód Python nyelven íródott, melynek széleskörű ismertségének (és a rendszer nyílt forráskódú voltának) köszönhetően bárki kiegészítheti a meglévő kódot.

A szakdolgozat felépítése

TODO

1. fejezet

A Jporta (5-10 oldal)

Az Irányítástechnika és Informatika Tanszéken 2009 és 2015 között aktívan használták a CPorta névre keresztelt, előbbi tulajdonságok jeletős részével rendelkező rendszert [5] a C és C++ tárgyak oktatásánál. A rendszer annak idején PHP nyelven íródott, karbantartása pedig nehézkessé vált. Emiatt

2. fejezet

Hallgatók értékelései

2.1. Értékelési rendszer (2-3 oldal)

Értékelési formák, stb..

2.2. Dinamikus mezők (3-5 oldal)

2.2.1. Dinamikus mezők egymásra hivatkozása

3. fejezet

Automatizált feladatkiértékelő modul

3.1. Megvalósult funkciók

3.1.1. Jogosultságok felülvizsgálata (2-3 oldal)

Acl, django permissions, stb...

3.1.2. Kód lefedettség ellenőrzés (2-3 oldal)

3.1.3. Feladatok (tesztek) importálása és exportálása (2-3 oldal)

3.1.4. Feladatok csoportosítása (2 oldal)

3.2. További lehetőségek

3.2.1. Plágiumkeresés (1-2 oldal)

3.2.2. Verziókezelő támogatás (1-2 oldal)

Összefoglalás

Köszönetnyilvánítás

Ábrák jegyzéke

Irodalomjegyzék

- [1] Dominique Thiébaud: Automatic evaluation of computer programs using Moodle's virtual programming lab (VPL) plug-in. *Journal of Computing Sciences in Colleges*, 30. évf. (June 2015) 6. sz., 145–151. p.
- [2] edX Inc.: About EdX. <https://www.edx.org/about-us>. Elérve: 2017. november 11.
- [3] edX Inc.: Year in Review: edX in 2016. <https://blog.edx.org/year-review-edx-2016?track=blog>. Elérve: 2017. november 11.
- [4] Free Software Foundation, Inc.: GNU General Public Licenc. <https://www.gnu.org/licenses/gpl.html>. Elérve: 2017. november 09.
- [5] Kálmán Viktor: Integrált feladatellenőrző és oktatórendszer. Diplomaterv (Budapesti Műszaki és Gazdaságtudományi Egyetem). 2016. december.
- [6] Moodle Pty Ltd: About moodle. https://docs.moodle.org/33/en/About_Moodle. Elérve: 2017. november 09.
- [7] Moodle Pty Ltd: Moodle Statistics. <https://moodle.net/stats/>. Elérve: 2017. november 09.
- [8] Virtual Programming Lab. <http://vpl.dis.ulpgc.es/>. Elérve: 2017. november 09.